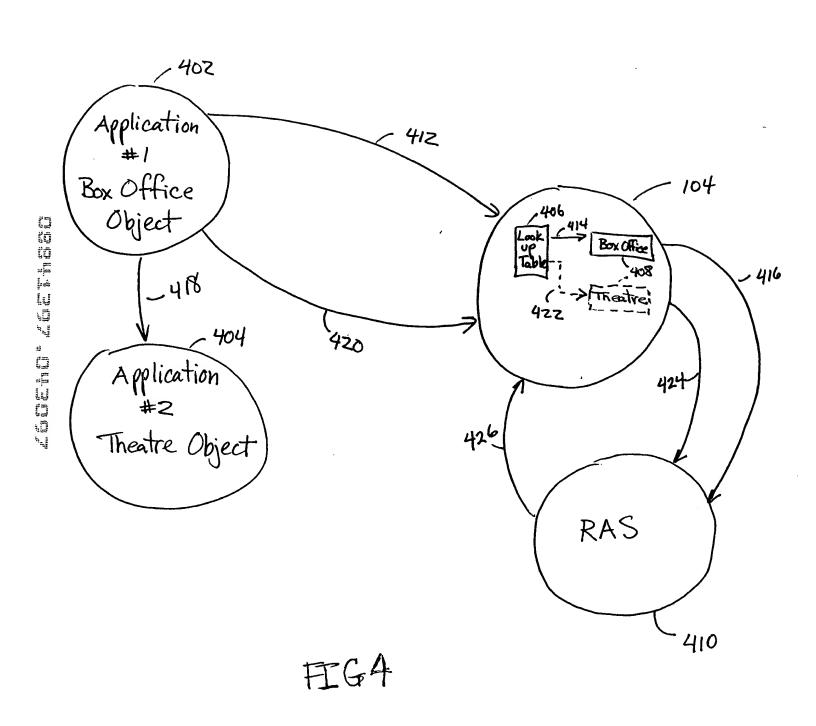
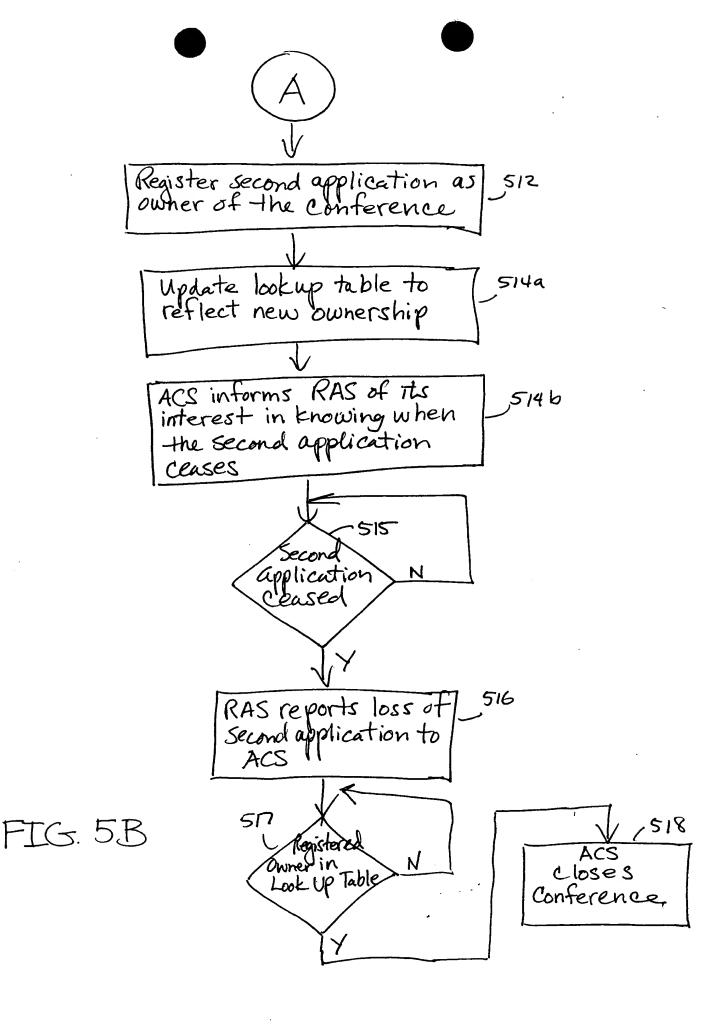


FIG. 3



500 First application Starts the conference 502 Register first application as owner of the conference 506a Enter owner of conference in lookup table in ACS ACS informs RAS of its 506b interest in knowing when first application cases *50*7 Sonference First application moves conference to a second 508 application Pass conference ownership to second application

FIG. 5A



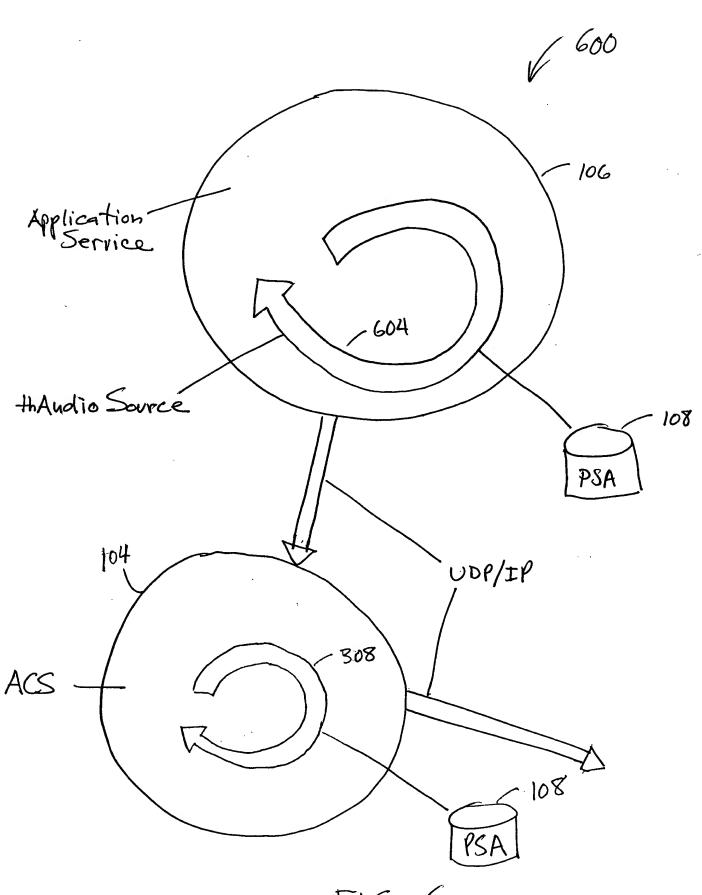


FIG. 6

National *Brand



		807	2 /804	1 806				
	Source	audio 1	audio z	audio 3				
	Target							
802 ~	Jaudiol	0.0	1.0	0.7	~ ~			
	Jaudioz	0.8	0.0	0.5				
80H	Jaudio3	6.5	6.5	0.0				
806			-					
	·				() (
	[audio] = audio] *0.0 + audioZ *10 + audio 3 *0.7 =]							
868	andioZ=andio1*0.8+andio2*0.0+andio3*0.5							
	andioz=a	udio1*0.8+	audio2*0.0+0	audio3*0.5				
	audio3=0	endial*6.5+	audio2*0.5+	audio3*00=	(

FIG. 8

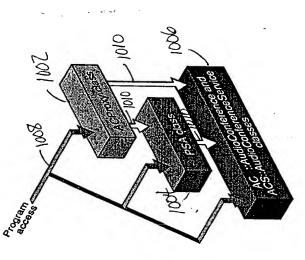
Receive audio data from STBs and PSAs

Mix audio dosto 9 for each target audio client for spatialized audio

Deliver actual andio 900 mixed data to the designated target audio client

FIG. 9A

910 Identify a decay factor for each source budio client 912 914 Determine the distance between the target audio client and each source-audioclient Extract weighted values from 716 the identified decay factor plots according to distance 918 Enter the weighted values into a mixtable for each source/taget and io Client pair 920 Calculate the mixed audid values for each target audio client 922 Refine the mixed audio values FIG. 9B



42-389 200 SHEET 42-399 200 RECYC 42-399 200 RECYC

ACProxy Class Methods ~ 1100

		Method	Description .
1102	~	ACProxy()	Opens the audio conference.
104	-	~ACProxy()	Cleans up and closes the audio conference.
1106	<u> </u>	AddClient()	Adds set-top audio client to the audio conference.
1108	-	AddPSA()	Adds a PSA audio client to the conference.
1110		Audios()	Lists audio client IDs in the conference.
1112	1	DemuteAudio()	Enables audio of the PSA.
1114	1	GetAudioLocation()	Locates X,Y coordinates of an audio client.
1116	1	GetConfInfo()	Retrieves audio conference information.
1118	_	MoveAudio()	Moves audio source to a specified location.
1120	_	MuteAudio()	Disables audio of the PSA.
1122		RegisterOwner()	Transfer owner of the audio conference using the object reference.
1124	سر	RegisterOwnerByName()	Transfer owner of the audio conference using the name of the application as exported by the Name Space.
1126		RemoveAudio()	Removes the audio client.
1128		UnregisterOwner()	Removes previous ownership of the audio conference.

PointSourceAudio Class Methods ~ 1200

Method	Description		
i202~PointSourceAudio()	Class constructor. Instantiates a PSA object.		
1204 ~PointSourceAudio()	Cleans up resources, closes the device used by a PSA.		
1266-Play()	Plays the audio source.		
1208n Stop()	Stops playing the audio source.		
12/0~ Pause()	Pauses the playing of the audio source.		
12/21 Resume()	Resumes playing the audio source.		

Fig. 12

1300

The ACS::AudioConferenceService methods

OpenConference() — creates an audio conference. 1302

CloseConference() — closes an audio conference. 1304

GetConferenceByTicket() — finds the conference name given the ticket number. 1306

ListConference() — returns a sequence of conference objects. 1308

HctAudioStat() — finds information about a set-top audio client.

1400 -	NewAudio() — creates an audio client for a given set-top box.
--------	---

1404 - • DeleteAudio() — deletes an audio client.

/406 - RegisterOwner() - registers conference owner by object reference

/408 - RegisterOwnerByName() — registers conference owner by export name

/4/0 • UnregisterOwner() — removes a registered conference ownership

/4/2 • SetMixOne() — sets the volume of the conversation between two audio clients.

/4/4 • SetMix() — sets the volume of the conversations between many audio client pairs.

 $/\frac{4}{6}$ • GetMixOne() — returns the volume setting between two audio client pairs.

1418 — • GetMix() — returns the volume settings between many audio client pairs.

• SetTimeOut() — sets a number of seconds of inactivity, after which the conference is closed automatically. It can be set to never close.

(2)2 • GetTimeOut() — finds the length in seconds of the timeout.

• AudioIds() — gets a list of audio IDs in a conference.

• AudioStat() — gets information about an audio client given the audio ID.

ConfStat() — gets information about an audio conference.

• HctAudioStat() — gets information about an audio client given the set-top ID.

Ping() — pings a given audio conference and returns an exception if the conference doesn't exist.

FIG. 14

42-391 50 SHEETS EYE-ENSE'S SOUAN 42-392 100 SHEETS EYE-ENSE'S SOUAN 42-392 100 SHEETS EYE-ENSE'S SOUAN 42-392 100 RECYCLED WHITE SSOUAN 42-393 200 RECYCLED WHITE SSOUAN

Add an audio client to the proxy audio conference

Locate the audio dient onscreen

FIG. 15

And the first that the first test of the first t

1600 1602 open an audio conference in an application 1604 Add an audio client 1606 Set the volume between audio clients 1808 Remove the audio client from the conference 1610 Close the audio conference

FIG. 16